الجامعة العربية الخاصة للعلوم والتكنولوجيا

كلية الهندسة المعلوماتية

السنة الثالثة

مقرر تحليل وتصميم الأنظمة – القسم النظري

الفصل الأول 2024/2023

المحاضرة الأولى والثانية

م. رنيم الحلبي

الفصل الأول

دورة حياة تطوير النظم Systems Development Life Cycle

أهداف الفصل

يهدف هذا الفصل إلى:

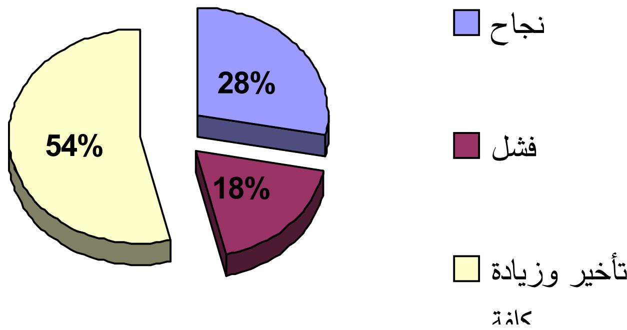
1) تعرّف مفهوم دورة حياة تطوير النظم بمراحلها الأربع وفهم هذه المراحل
2) تعرّف أصناف المنهجيات المختلفة المستخدمة في تطوير النظم، وكيفية اختيار المنهجية المناسبة
3) التآلف مع المهارات المختلفة اللازمة لفريق التطوير وأدوار العاملين فيه

مقدمة

تعريف دورة حياة تطوير النظم Systems Development Life Cycle (SDLC) :

بأنها الإجراء الذي يجري من خلاله فهم كيف يمكن لنظام برمجي أن يدعم حاجة العمل، وتصميم هذا النظام، ثم بناؤه، وتسليمه للمستخدم. وقد سبق لك عزيزي الطالب أن أخذت أكثر من مقرر في البرمجة وقمت بكتابة بعض البرامج، لـ ذا، يخيل إليك بإن الأمر بسيط.

غير أنه في الحقيقة ليس كذلك. فقد بينت إحدى الدراسات التي أجراها Standish Group عام 2004 أن 28% فق ط م ن المشاريع البرمجية يكتب لها النجاح، في حين أن 18% من المشاريع تلغى قبل إنجازها. أما ما تبقى من الم شاريع ، فهي مشاريع تسلّم لمستخدميها مع تأخير كبير في أزمن ة التسليم وزيادة واضحة في الكلفة عما هو متف ق علي ه، وفوق كل هذا فهي لا تحقق س وى ج زء ص غير م ن الوظائف المطلوبة.



قد تعتقد عزيزي الطالب أن هذا يحصل "للآخرين" ولا يمكن أن يحصل لك! غير أن الواقع يشير إلى حدوث هذا الأمر في معظم الشركات! ويمكنك أن تطلّع على العديد من قصص الفشل[*] في المشاريع البرمجية.

في الحقيقة لا يوجد حل سحري يضمن نجاح البرمجيات المطورة، غير أننا في هذا المقرر سنعرض عدداً م ن المب ادئ الأساسية والتقنيات العملية التي نأمل أن يؤدي تطبيقها إلى ارتفاع احتمال نجاح البرمجيات.

إن أهم شخص في دورة حياة تطوير النظم هو محلل النظام.

يقوم محلل النظام بتحليل واقع العمل وتحديد مواضع التحسينات، ويصمم نظام المعلومات المناسب لتنفيذ هذه التح سينات. وتعتبر مهنة محلل النظام من أكثر المهن تحدياً وإثارة، إذ يعمل المحلل مع عدة أصناف من العاملين ويتعلم م نهم كي ف يقومون بعملهم. كما يعمل المحلل ضمن فريق من المحللين والمبرمجين وغيرهم لأداء مهمة مشتركة وهي بن اء النظ ام البرمجي. وكم تكون سعادة المبرمج كبيرة عندما يرى كيف استطاع النظام البرمجي الذي شارك في صنعه أن يغير واقع العمل في المؤسسة ويحدث فرقاً نوعياً في آليات العمل.

من الضروري أن نشير هنا إلى أن هدف المحلل يجب أن ينصب على بناء نظام يعطي قيمة م ضافة للمؤس سة الت ي سيوضع فيها مثل زيادة أرباحها لا أن يصنع برمجية رائعة. وهنالك العديد من البرمجيات التي فشلت لأن هدف المحل ل فيها كان بناء نظام برمجي رائع دون التركيز على فهم كيف سيدعم هذا النظام أهداف المؤسسة وإجراءات عملها الحالي ة ونظم المعلومات الأخرى التي تمتلكها.

فالاستثمار في شراء نظام معلومات جديد يشبه الاستثمار في شراء آلة جديدة، بمعنى أن الهدف ليس هو اقتناء النظام أو

الآلة إذ أنهما ليسا سوى وسائل لتحقيق غايات وأهداف المؤسسة أما الهدف الحقيقي فهو تمكين المؤسسة من أداء عملها بطريقة أفضل مما سينعكس على زيادة أرباحها ويخدم مكوناتها بفعالية أكبر .

يهدف هذا المقرر عزيزي الطالب إلى تعريفك بالمفاهيم الأساسية التي يحتاجها محلل النظم، وإلى توجيهك عبر المراحل التي تستطيع من خلالها بناء النظم البرمجية.

سنتناول في هذا الفصل دورة حياة تطوير النظم (SDLC) التي تمر بها نظم المعلومات. تتبع جميع نظم المعلومات دورة الحياة نفسها رغم اختلافها في المقاربة المستخدمة في كل مرحلة من مراحل دورة الحياة. تبين الفقرة التالية مراحل دورة الحياة ويليها عرض لثلاثة أنماط مختلفة من منهجيات التطوير.

دورة حياة تطوير النظم (SDLC)

إن بناء نظام معلومات يشبه من عدة وجوه بناء منزل. إذاً، لنتذكر معاً مراحل بناء المنزل تمهيداً لعرض مراحل بناء النظام البرمجي.

أولاً:	يبدأ بناء المنزل بفكرة أولية.
ثانياً:	يجري تحويل هذه الفكرة إلى رسم مبسط يعرض على الزبون. يجري تفصيل هذا الرسم الأولي وتحسينه عبر عدد من الرسومات التي يلي كل منها سابقه ويزيد عليه في التفاصيل إلى أن يرى الزبون أن الرسم المعروض أمامه يعبر عن الصورة التي يريدها لمنزله.
ثالثاً:	يقوم المهندس بوضع عدد من المخططات التفصيلية (التي تبين أنواع صنابير المياه، مواضع مآخذ الهاتف ...الخ)
رابعاً:	يتم بناء المنزل اعتماداً على هذه المخططات، وكثيراً ما يقرر الزبون القيام ببعض التعديلات أثناء عملية البناء.

دورة حياة تطوير النظم (SDLC)

إن بناء النظم البرمجية يمر بالمراحل الأربع آنفة الذكر وهي:

- التخطيط Planning
- التحليل Analysis
- التصميم Design
- التنجيز Implementation

تتألف كل مرحلة من هذه المراحل من عدة خطوات تعتمد كل منها على مجموعة من التقنيات، وتنتج وثائق مخصصة تعطي في مجموعها فهماً للنظام.

يعرض الجدول التالي ملخصاً يبين تعريف هذه المراحل و خطواتها والتقنيات المستخدمة في كل منها والنواتج التي تنتج عن كل منها. تذكر عزيزي الطالب أننا سنركز في هذا المقرر بشكل أساسي على التحليل والتصميم.

المرحلة	الخطوات	التقنيات	النواتج
التخطيط Planning: يركز على الأسئلة التالية: – لماذا نبني النظام؟ –كيف تكون بنية المشروع؟ الخرج الأساسي: – طلب النظام مع دراسة جدوى – خطة المشروع	تحديد مدى الحاجة تحليل الجدوى تنظيم خطة عمل تزويد المشروع بالعاملين إدارة المشروع ومتابعته	تعريف المشروع الجدوى الفنية الجدوى الاقتصادية الجدوى التنظيمية تقدير الوقت اللازم تحديد المهام البنية المجزأة للأعمال مخطط PERT مخطط Gantt توظيف فريق العمل مستودع CASE[2] معايير توثيق إدارة المخاطر	طلب النظام دراسة الجدوى خطة المشروع –خطة العمل –خطة التوظيف –قائمة المعايير –تقييم المخاطرة
التحليل Analysis: يركز على الأسئلة التالية: من المعني بالنظام، ماذا ومتى وأين؟	تطوير استراتيجية تحليل تحديد متطلبات الأعمال	أتمتة إجرائيات العمل تحسين إجرائيات العمل إعادة هندسة إجرائيات العمل	مقترح للنظام – تحديد المتطلبات

[2] Computer Aided Software Engineering أداة من أدوات هندسة البرمجيات بمساعدة الحاسوب تكون عبارة عن بيئة متكاملة يمكنها أن تغطي كامل مراحل التطوير البرمجي

الخرج الأساسي: - مقترح للنظام	بناء حالات الاستخدام نمذجة الإجرائيات نمذجة المعطيات	مقابلات جلسات JAD[3] استبيانات تحليل الوثائق المراقبة تحليل حالات الاستخدام مخططات تدفق المعطيات نمذجة العلاقات بين الكيانات التحويل إلى علاقات نظامية	- حالات الاستخدام - نماذج الإجرائيات - نماذج المعطيات
التصميم Design: يركز على الأسئلة التالية: كيف سيعمل هذا النظام؟ الخرج الأساسي: - توصيف النظام	تصميم النظام المادي تصميم البنيان تصميم الواجهات تصميم البرمجيات تصميم قواعد المعطيات والملفات	استراتيجية التصميم تصميم البنيان انتقاء البرمجيات والتجهيزات سيناريوهات الاستخدام بنية الواجهات معايير الواجهات بروتوتايب الواجهات تقييم الواجهات رسم مخططات تدفق المعطيات مخططات بنية البرنامج توصيف البرنامج انتقاء صيغة المعطيات نمذجة العلاقات بين الكيانات تحسين الأداء تقدير الحجوم	مصفوفة البدائل توصيف النظام - تقرير البنيان -توصيف البرمجيات والتجهيزات - تصميم الواجهات -النموذج الفيزيائي للإجراءات - تصميم البرنامج -توصيف قواعد المعطيات والملفات -النموذج الفيزيائي للمعطيات

[3] Joint Application Development التطوير المشترك للتطبيقات وهي تقنية تقضي بمشاركة كل من المحلل والزبون في التطوير

التنجيز Implementation: يركز على الأسئلة التالية:	بناء النظام تنزيل النظام صيانة النظام ما بعد التنجيز	البرمجة الاختبارات البرمجية اختبارات الأداء اختيار استراتيجية التحول التدريب انتقاء الدعم صيانة النظام تقييم المشروع فحوص ما بعد التنجيز	خطة الاختبارات البرامج التوثيق خطة التهجير – خطة التحول – خطة أعمال الطوارئ – خطة التدريب خطة الدعم تقرير المشاكل طلب التغيير تقرير فحوص ما بعد التنجيز

دورة حياة تطوير النظم (SDLC)

لنأخذ مثلاً عملية انتساب طالب إلى جامعة، ولنحاول تحسس معنى هذه المراحل والخطوات والتقنيات والنواتج في هذا المثال.

للانتساب إلى جامعة ما، يمر الطالب بعدة مراحل: جمع المعلومات عن الجامعات، ثم التقدم بطلب إلى جامعة ما، وتنتهي العملية أخيراً بالقبول في الجامعة.

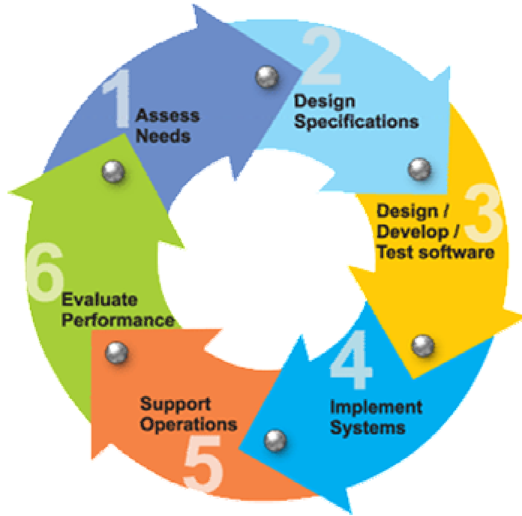


تتألف كل مرحلة من هذه المراحل من عدة خطوات، فمرحلة جمع المعلومات مثلاً تتضمن الخطوات التالية: البحث عن الجامعات وطلب المعلومات عن الاختصاصات المتاحة في هذه الجامعات، وقراءة النشرات الإعلانية.

يستخدم الطالب أثناء ذلك تقنيات مختلفة (كالبحث على الإنترنت) في مرحلة طلب المعلومات عن الاختصاصات وينشئ نواتج (مثل مقارنة الجوانب المختلفة للجامعات).

يقترح الجدول السابق أن تطبق مراحل دورة حياة تطوير النظم وخطواتها تطبيقاً يعتمد على مسار منطقي يبدأ من بداي ة المراحل وينتهي بنهاياتها. يكون هذا المسار مناسباً في بعض المشاريع البرمجية، غير أنه ف ي العدي د م ن الم شاريع الأخرى، قد تقوم فرق العمل المختلفة بالتنقل بين المراحل تتابعياً أو تزايدياً أو تكرارياً أو بطرق أخرى.

نعرض في هذه الفقرة تعريف المراحل وخطواتها وبعض التقنيات المستخدمة فيها، وسنعرض في فقرات أخ رى نم اذج التنقل بين هذه المراحل.



من المهم أن تعلم عزيزي الطالب أن SDLC ه ي إج راء يتمي ز بتدرجه في زيادة التفصيل. فالنواتج التي تنتج عن مرحلة التحلي ل تعطي فكرة عامة عن شكل النظام الجديد، وتستخدم كمدخل لمرحلة التصميم. يجري خلال التصميم تفصيل هذه النواتج لإنتاج ن واتج تصف بمصطلحات أدق كيف سيجري بناء النظام. ت ستخدم ن واتج مرحلة التصميم بدورها في مرحلة التنجيز لإنتاج النظ ام الفعل ي. وهكذا، نرى أن كل مرحلة تستند إل ى ن واتج المرحل ة ال سابقة وتُطورها.

سنعرض في الفقرات التالية إلى كل مرحلة من مراحل دورة الحياة بإيجاز.

المرحلة الأولى: التخطيط

تهدف مرحلة التخطيط إلى فهم المبررات التي تدعونا إلى بناء نظام معلومات، كما تحدد كيف سيسعى فريق المشروع إلى بنائه. تتألف هذه المرحلة من خطوتين أساسيتين:

❶ الخطوة الأولى: إقلاع المشروع Project Initiation : حيث يجري تحديد الفائدة المرجوة من النظام في المؤسسة، هل سيرفع النظام قيمة العائدات أم سيخفض النفقات؟

وقد لوحظ أنه في معظم الحالات كان طلب النظام الجديد يصدر من خارج مديرية المعلوماتية. يكون *طلب النظام* عب ارة عن وثيقة تتضمن ملخصاً يعبر عن الحاجة إلى نظام برمجي وتشرح كيف يمكن للنظام المطلوب أن يحسن أداء العم ل. عندما تتلقى مديرية المعلوماتية هذا الطلب، فإنها تعمل بشكل وثيق مع القسم الذي صدر عنه الطلب (والذي يسمى مم ول المشروع project sponsor) لإجراء تحليل للجدوى.

يركز تحليل الجدوى على دراسة النقاط الأساسية الثلاث التالية:

- الجدوى التقنية (هل يمكننا بناء النظام؟)
- الجدوى الاقتصادية (هل سيقدم النظام قيمة مضافة؟)
- الجدوى التنظيمية (في حال تم بناء النظام، هل سيجري استخدامه؟)

يقدم طلب النظام وتحليل الجدوى إلى لجنة الإشراف على نظم المعلومات لتقرر فيما إذا كان م ن الممك ن تبن ي ه ذا

المشروع.

❷ **إدارة المشروع Project Management :**

يقوم مدير المشروع بوضع خطة عمل Work Plan ، كما يحدد فريق العمل ويوفر التقنيات اللازمة لمساعدته في إدارة المشروع. في نهاية هذه الخطوة يقدم مدير المشروع خطة عمل لمشروعه.

المرحلة الثانية: التحليل

تسمح مرحلة التحليل بالإجابة عن الأسئلة التالية:

- *من* الذي سيستخدم النظام؟
- *ما الذي* سيفعله النظام؟
- *متى وأين* سيستخدم النظام؟

في هذه المرحلة يقوم فريق المشروع بالتحري عن وجود أي نظام حالي ويحدد إمكانات التطوير الممكنة ويطور مفهوماً للنظام الجديد.

تتألف هذه المرحلة من ثلاث خطوات:

❶ **تطوير استراتيجية للتحليل Analysis Strategy :** لترشيد جهود الفريق. تتضمن هذه الاستراتيجية عادة تحليلاً للنظام الحالي (as-is system) ومشكلاته، وتصوراً للنظام الجديد (to-be system).

❷ **جمع المتطلبات Requirement Gathering:**. يقود تحليل المعلومات التي جمعت إلى تشكيل مفهوم واضح عن النظام الجديد. يستخدم هذا المفهوم لصنع مجموعة من نماذج التحليل

❸ **مقترح النظام System Proposal:** تجمع التحليلات التي تم الحصول عليها مع مفهوم النظام والنماذج في وثيقة تسمى وتقدم إلى ممول المشروع وإلى المعنيين باتخاذ القرار حول بناء النظام.

> تعتبر وثيقة مقترح النظام الناتج الأول الذي يصف متطلبات النظام المطلوب. كما ينتج عن هذه المرحلة تصميماً أولياً للنظام الجديد.

المرحلة الثالثة: التصميم

في هذه المرحلة يتم اتخاذ القرارات حول *كيفية* عمل النظام الجديد معبرين عن ذلك بالتجهيزات والبرمجيات والبنية الشبكية اللازمة للمؤسسة وواجهات التخاطب مع المستخدم والاستمارات والتقارير التي ستستخدم، إضافة إلى البرامج وقواعد المعطيات والملفات التي يحتاجها النظام. تتألف هذه المرحلة من أربع خطوات:

❶ **تطوير استراتيجية للتصميم Design Strategy:** توضح هذه الاستراتيجية إذا ما كان سيجري تطوير النظام داخل المؤسسة أم خارجها.

❷ **تطوير تصميم البنيان الخاص بالنظام Architecture Design:** ويتضمن وصفاً للتجهيزات والبرمجيات والبنية الشبكية التي ستستخدم، إضافة إلى تصميم الواجهات حيث تحدد كيفية التخاطب مع المستخدم والاستمارات

والتقارير اللازمة.

❸ **توصيف قواعد المعطيات والملفات التي يحتاجها النظ ام databases and file specifications :** وهنا يجدد بدقة المعطيات التي يجب تخزينها ومكان تخزينها.

❹ **تصميم البرنامج Program Design:** ويتضمن تحديد البرامج التي يجب كتابتها، والتوصيف الدقيق لعمل هذه البرامج.

المرحلة الرابعة: التنجيز

يجري في هذه المرحلة بناء النظام أو شراؤه (في حالة تقرير شراء حزمة برمجيات جاهزة). تكون هذه المرحل ة ع ادة أطول المراحل وأكثرها كلفة، وهي تتألف من ثلاث خطوات:

❶ **إنشاء النظام System Construction:** يجري بناء النظام واختباره للتأكد من أدائه العمل كما جرى تصميمه. وتعتبر عملية الاختبار من أكثر العمليات كلفة.

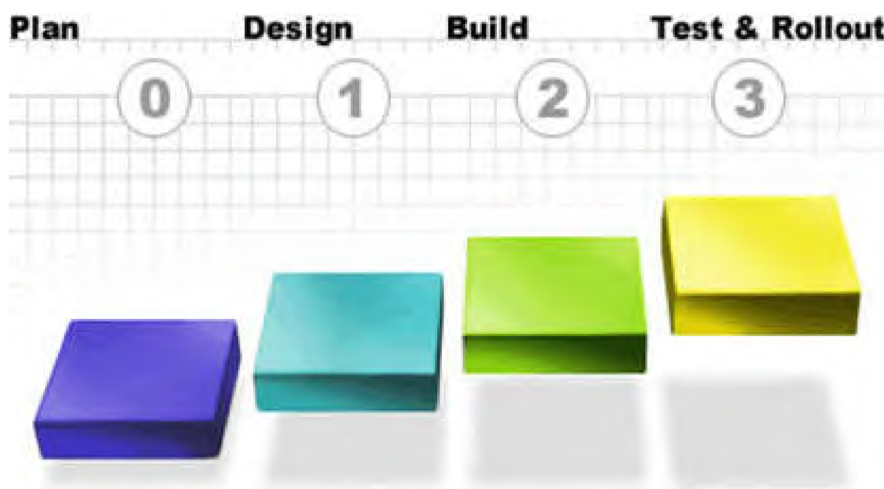
❷ **التثبيت عند الزبون Installation:** وهنا يجري وضع النظام لدى الزبون تدريجياً أو كلياً حسب إجرائية يتف ق عليها. وتتضمن هذه المرحلة وضع خطة لتدريب المستخدمين training plan على النظام الجديد.

❸ **وضع خطة لدعم الزبون Support Plan:** تتضمن إجراء مراجعة للنظام في مرحل ة الاس تثمار، وتحدي د التعديلات الصغيرة أو الكبيرة التي يحتاجها النظام.

منهجيات تطوير النظم

تُعرّف المنهجية بأنها المقاربة المستخدمة أثناء وضع دورة حياة تطوير النظم موضع التنفيذ. بمعنى أنها تتألف من لائح ة من الخطوات والنواتج.

وتختلف المنهجيات بعضها عن بعض بحسب تركيزها على إجرائيات العمل أو على المعطيات التي تدعم العمل، وبحسب ترتيبها وتأكيدها على كل مرحلة من مراحل دورة حياة تطوير النظم.



منهجيات تطوير النظم

سنعرض فيما يلي للأصناف المختلفة التي تنتظم فيها المنهجيات تبعاً لمحاور اهتمامها(انقر على كل منها).

.1	**المنهجيات المتمركزة حول الإجرائيات Process-centered Methodologies**	يركز هذا الصنف من المنهجيات على البدء بتعريف الأنشطة المتعلقة بالنظام (أي الإجرائيات). يستخدم المحللون نماذج الإجرائيات، حيث يمثلون المفاهيم الموجودة في النظام كمجموعة من الإجرائيات التي تتدفق المعلومات فيما بينها.
.2	**المنهجيات المتمركزة حول المعطيات Data-centered Methodologies**	يركز هذا الصنف من المنهجيات على البدء بتعريف محتويات حاويات خزن المعلومات وكيفية تنظيم هذه الحاويات. يستخدم المحللون نماذج المعطيات، حيث يمثلون المفاهيم الموجودة في النظام كمجموعة من المعطيات التي تنظم في بنى محددة.
.3	**المنهجيات الغرضية التوجه Object-oriented Methodologies**	تسعى هذه المنهجيات إلى تحقيق التوازن بين التركيز على المعطيات والتركيز على الإجرائيات. وتستخدم لغة النمذجة الموحدة Unified Modeling Language (UML) لوصف مفاهيم النظام باعتبارها مجموعة من الأغراض التي تتضمن في الوقت ذاته المعطيات والإجرائيات.

كما يعتبر ترتيب مراحل دورة حياة تطوير النظم والوقت الذي يكرس لكل منها من العوامل الهامة التي تساعد على تصنيف المنهجيات. وسنعرض فيما يلي ثلاثة أصناف رئيسية من منهجيات تطوير النظم التي تطورت مع الزمن.

التصميم البنيوي Structured Design

تتبنى منهجيات التصميم البنيوي مقاربة صورية لدورة حياة تطوير النظم تعتمد سياسة الخطوة خطوة، بمعنى أن يجري الانتقال بشكل منطقي من مرحلة إلى المرحلة التي تليها.

وتعتبر هذه المنهجيات الرائدة في إدخال النمذجة الصورية وتقنيات رسم المخططات لوصف إجرائيات العمل الأساسية في النظام والمعطيات التي تدعمها. وتنقسم إلى نوعين: التطوير الشلالي والتطوير على التوازي.

التطوير الشلالي Waterfall Development

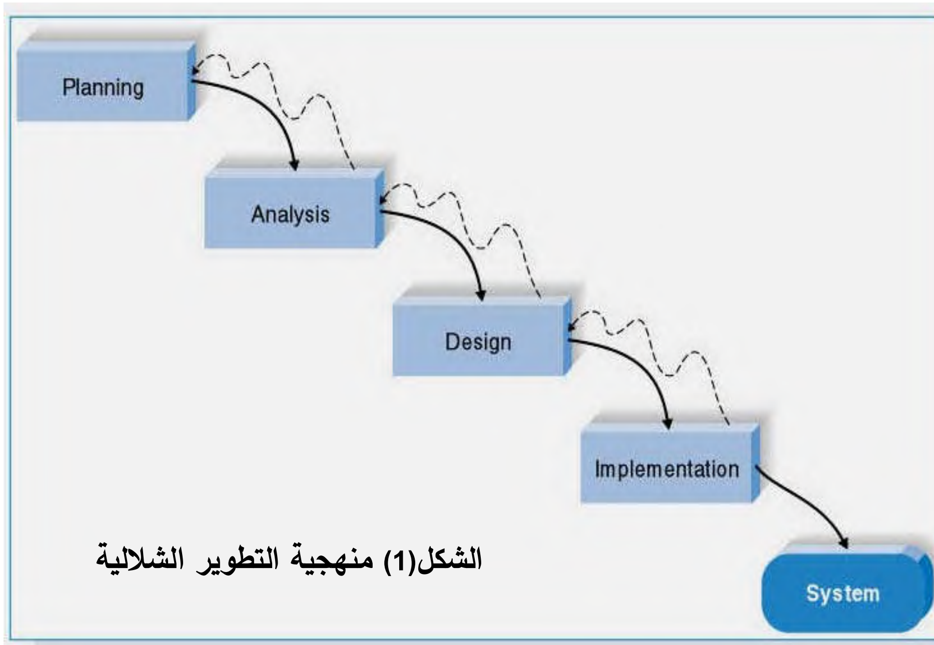
في المنهجيات المعتمدة على التطوير الشلالي، ينتقل المحللون والمستخدمون انتقالاً تتابعياً من مرحلة لأخرى.

وتتميز هذه المنهجيات بالميزات التالية:

- تُحدد متطلبات النظام قبل البدء بالبرمجة بوقت طويل.
- مع تقدم المشروع تضعف إمكانية إجراء تعديلات على المتطلبات.

أما مساوئ هذه المنهجيات فهي التالية:

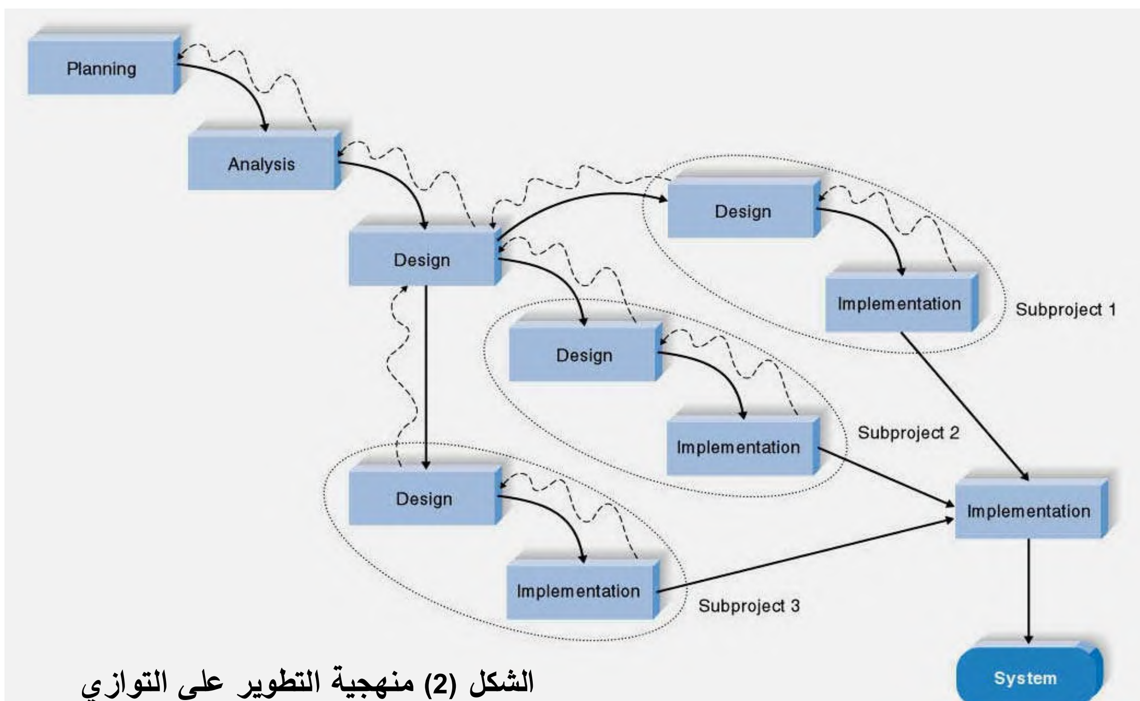
- يجب أن ينتهي التصميم تماماً قبل البدء بالبرمجة.
- يمر وقت طويل بين طلب النظام وتسليمه.



الشكل(1) منهجية التطوير الشلالية

التطوير على التوازي Parallel Development

تحاول المنهجيات التي تعتمد التطوير على التوازي أن تعالج موضوع الفترة الزمنية الطويلة التي تمر بين طلب النظام وتسليمه. فبدلاً من القيام بالتصميم كاملاً ثم الانتقال إلى التنجيز (كما في التطوير الشلالي)، يوضع تصميم عام للنظام ككل، ثم يقسم المشروع إلى عدد من المشاريع الفرعية المستقلة التي يمكن تصميم كل منها وتنجيزه على التوازي مع المشاريع الفرعية الأخرى (انظر الشكل (2)).



الشكل (2) منهجية التطوير على التوازي

منهجيات تطوير النظم

التطوير السريع للتطبيقات Rapid Application Development (RAD)

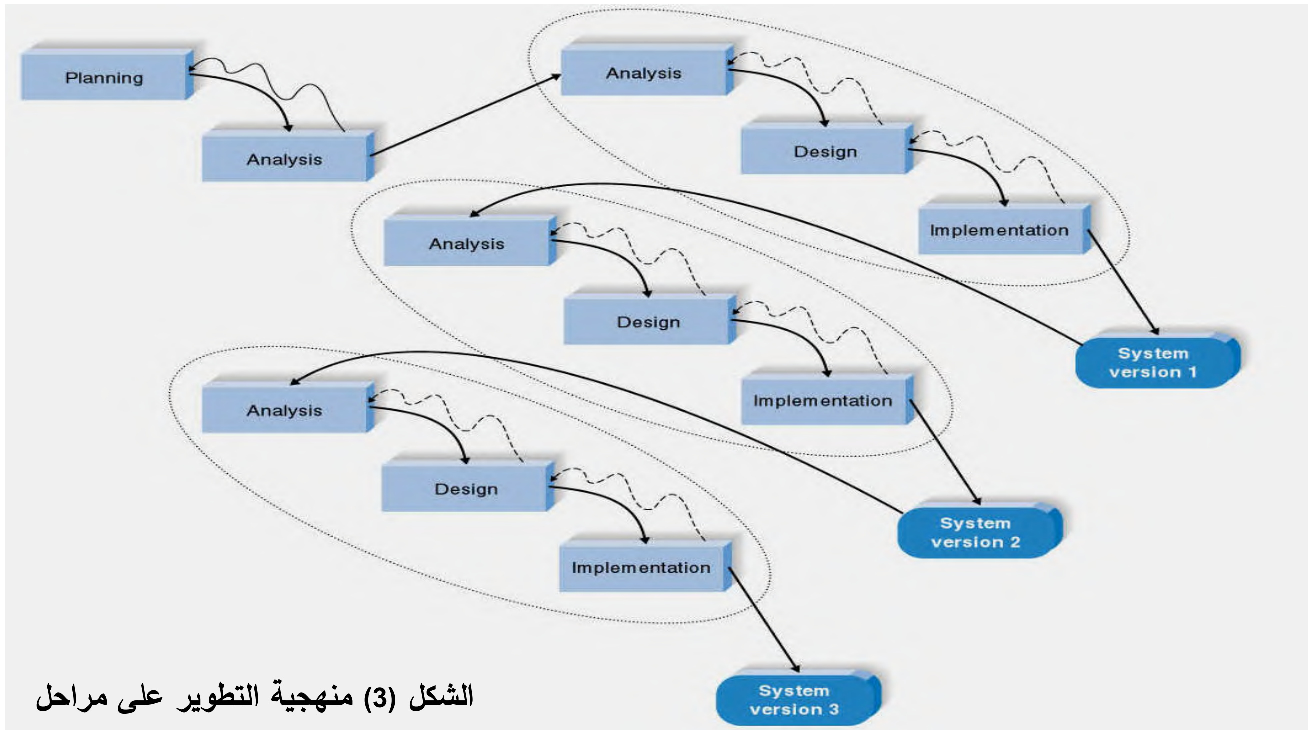
ظهرت المنهجيات المعتمدة على التطوير السريع للتطبيقات للتغلب على نقطتي الضعف المذكورتين آنفاً ف ي منهجي ات التصميم البنيوي. لتحقيق هذا الهدف، تنسق المنهجيات المعتمدة على RAD بين مراحل دورة حياة تطوير النظام للحصول على أجزاء من النظام بسرعة وتضعها بين يدي المستخدم. إن حصول المستخدم على أجزاء من النظام في وق ت مبك ر يتيح له فهماً أفضل للنظام مما يجعله يقترح بعض التعديلات التي تجعل النظام أكثر تلبية لاحتياجاته.

تنصح معظم المنهجيات المعتمدة على التطوير السريع للتطبيقات أن يستخدم المحللون تقنيات مخصصة وأدوات حاسوبية خاصة لتسريع مراحل التحليل والتصميم والتنجيز، مثل أدوات هندسة البرمجيات بمعون ة الحاس ب CASE، وجل سات JAD[4]، ولغات البرمجة المرئية (مثل Visual Basic.Net).

ومع ذلك تبقى هناك في المنهجيات المعتمدة على التطوير السريع للتطبيقات مشكلة خفية تكمن في إدارة توقعات الزبون. فمع زيادة سرعة تطوير النظام يزداد فهم الزبون للتقانات المستخدمة وتزداد طلباته وتوقعاته من النظ ام، مم ا يجع ل متطلبات النظام تتضخم بشكل كبير أثناء المشروع.

التطوير على مراحل Phased Development

تعتمد هذه المنهجيات على تجزئة النظام الكلي إلى سلسلة من الإصدارات التي يجري تطويرها تتابعياً. ففي مرحلة التحليل يجري تحديد المفهوم الكلي للنظام، ثم يقوم فريق المشروع والمستخدمون والممول بتصنيف المتطلبات في سل سلة م ن الإصدارات المتتابعة. تشكل المتطلبات الأساسية والأكثر أهمية الإصدار الأول.



الشكل (3) منهجية التطوير على مراحل

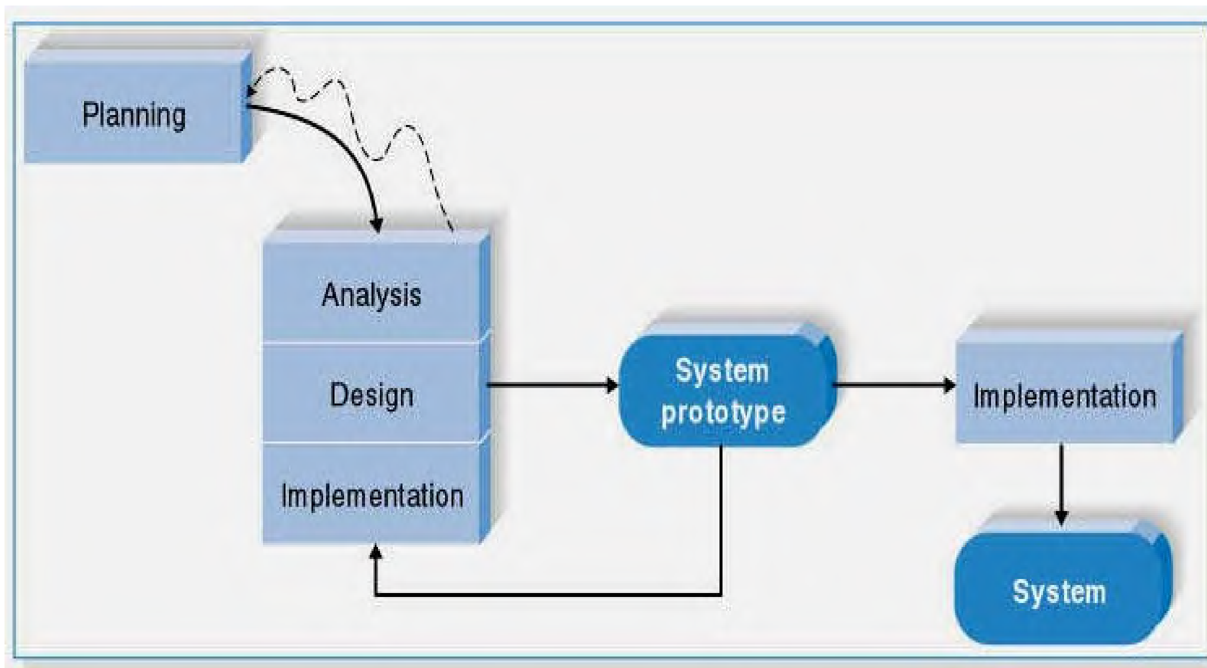
4 Joint Application Development التطوير المشترك للتطبيقات وهي تقنية تقضي بمشاركة كل من المحلل والزبون في التطوير

ننطلق من مجموعة المتطلبات هذه، وبعد تحليلها بدقة ننتقل إلى تصميها وتنجيزها فنحصل على الإصدار الأول من النظام . نكرر الأمر نفسه عدة مرات، بحيث أننا بعد انتهاء كل إصدار نبدأ العمل على الإصدار التالي الذي ينطلق من مجموعة المتطلبات السابقة مضافاً إليها الأفكار الجديدة التي يأتي بها المستخدم بعد تجربته مع الإصدار السابق (انظر الشكل (3) ، انقر على الشكل لتحصل على النسخة المكبرة ثم مرر المؤشر على الصورة لترى التوضيح).

النمذجة الأولية Prototyping

في هذه المنهجيات، يجري العمل في مراحل التحليل والتصميم والتنجيز بشكل تسايري. تُؤدّى هذه المراحل الثلاثة ضمن حلقة تكرارية إلى أن يتم إنجاز كامل النظام.

في مثل هذه المنهجيات نبدأ العمل بإجراء تحليل وتصميم أساسيين، ثم نقوم مباشرة ببناء النم وذج الأول ي للنظ ام. إن النموذج الأولي برنامج "سريع وفج" يعتبر إصداراً أولاً مصغراً عن النظام، ويمتلك عدداً قليلاً من الوظ ائف وال صفات المطلوبة من النظام (انظر الشكل (4)، ومرر المؤشر على الصورة لترى التوضيح). يشكل هذا الإص دار الج زء الأول الذي سيستخدمه الزبون، ويعرَض عادة على المستخدم والممول لأخذ ملاحظ اتهم وردود أفع الهم. بن اء عل ى ه ذه الملاحظات يجري العمل على نموذج فيه المزيد من الوظائف والصفات المطلوبة، وتكرر العملي ة ال سابقة إل ى ح ين الحصول على النظام بكامل مواصفاته.



الشكل (4) المنهجية المعتمدة على النمذجة الأولية

وتتميز المنهجيات المعتمدة على النموذج الأولي بأنها توفر للمستخدم نظاماً يمكنه التفاعل معه وإن لم يكن ه ذا النظ ام جاهزاً للاستخدام الفعلي.

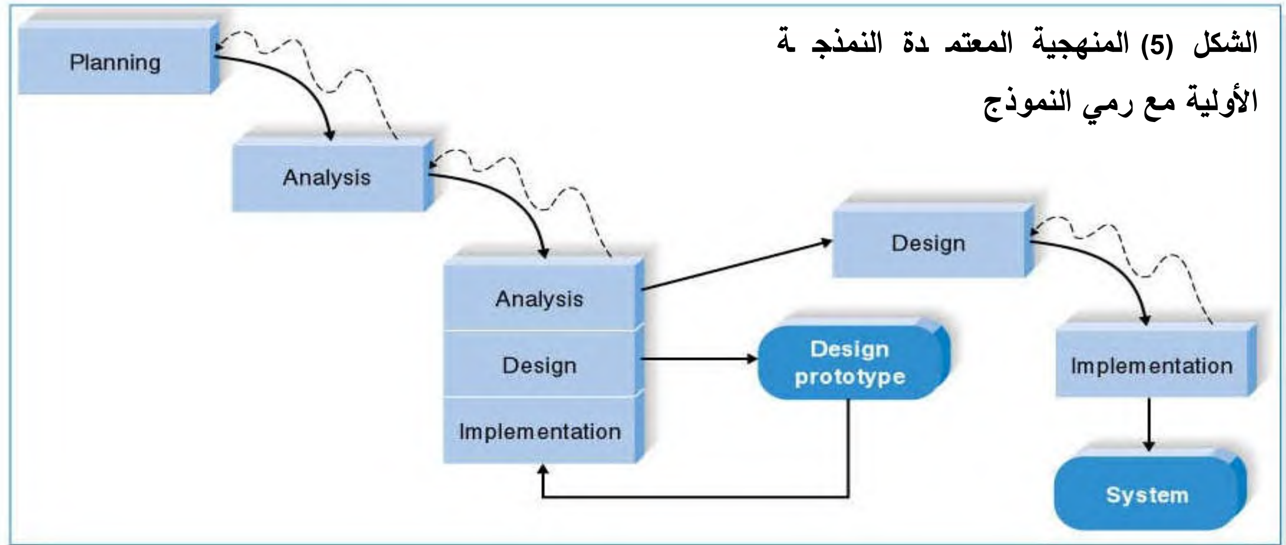
أما مساوئ هذه المنهجيات فهي أنها تؤدي في غالب الأحيان وبسبب كثرة التعديلات التي تطرأ على النموذج الأولي إل ى الحصول على تصميم سيء للنظام.

النمذجة الأولية مع رمي النموذج Throwaway Prototyping

تشبه هذه المنهجيات تلك التي أوردناها في الفقرة السابقة (المنهجيات المعتمدة على النموذج الأولي) في أنها تعتمد عل ى صنع نماذج أولية، غير أنها تختلف عنها في أنه يجري صنع النماذج الأولية في موضع مختلف من دورة الحياة.

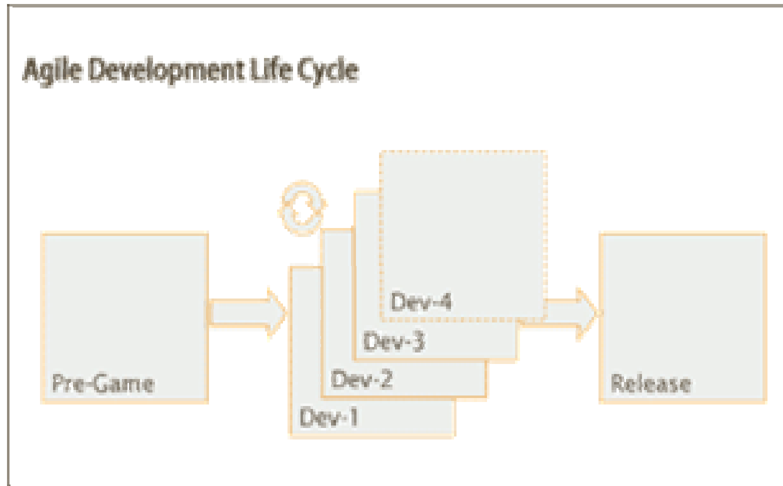
تقوم هذه النماذج بدور مختلف عن مثيلاتها التي أوردناها في الفقرة السابقة كما يكون مظهرها مختلفاً كلياً. يقوم المحللون في هذه المنهجيات بإجراء تحليل عميق نسبياً يتم خلاله جمع المعلومات تطوير أفكار حول مفهوم النظام. قد تكون بعض خصائص النظام التي يطلبها المستخدم غير واضحة أو خيالية أو تمثل تحدياً تقانياً، وعليه يجري تجريب كل م ن ه ذه الطروحات عبر بناء نموذج أولي تصميمي design prototype.

لا يعتبر هذا النموذج نظاماً لأنه في الحقيقة يمثل ذلك الجزء من النظام الذي يحتاج إلى تفصيل، وهو يتضمن التفاصـ يل التي تسمح بفهم الطرح الذي يجري سبره فقط (انظر الشكل (5) ومرر المؤشر عليه لترى التوضيح).



الشكل (5) المنهجية المعتمـ دة النمذجـ ة الأولية مع رمي النموذج

التطوير الرشيق Agile Development



أخذت هذه المنهجيات بالظهور حـ ديثاً. وه ي تركز بشكل كبير على عملية البرمجة، وتمتل ك عدداً قليلاً من القواعد والممارسات مما يجع ل اتباعها سهلاً.

تهدف هذه المنهجيات إلى الانسياب عب ر دورة حياة التطوير البرمجي بإلغاء الكثير من الحمل الإضافي الذي ين تج ع ن عملي ات النمذج ة والتوثيق مما يؤدي إلى ت وفير الوق ت ال ذي تستغرقه هذه العمليات. وبدلاً عن ذلك تركز المشاريع على تطوير بسيط وتكراري للتطبيقات.

ومن الأمثلة على هذه المنهجيات نورد البرمجة الحدية Extreme Programming (XP).

البرمجة الحدية (Extreme Programming (XP

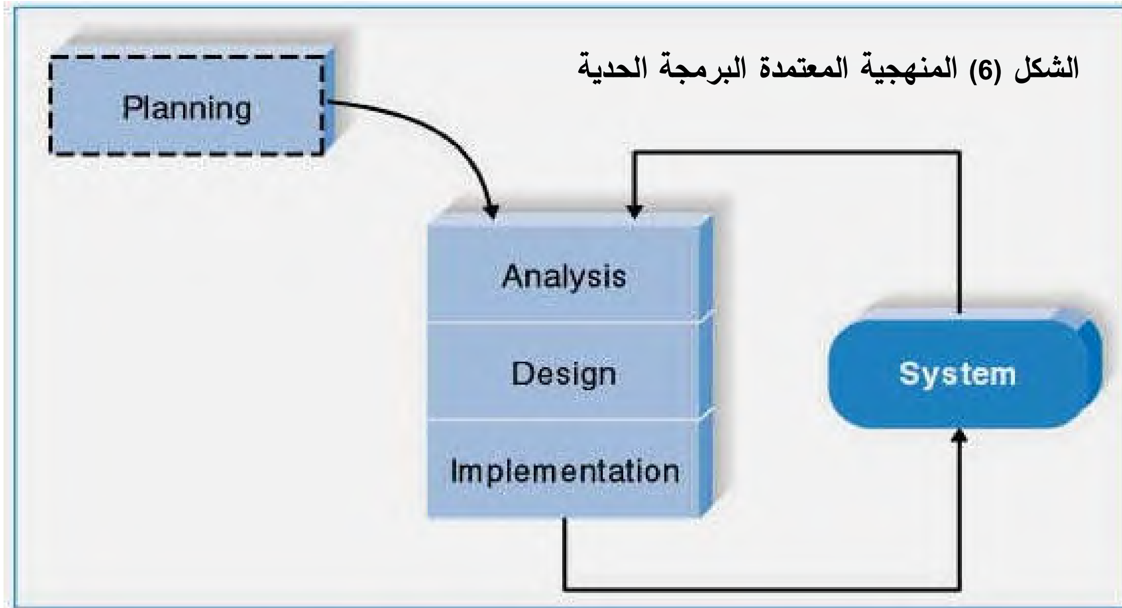
تقوم هذه المنهجية على القيم الأربعة التالية:

- **التواصل:** يجب أن يقدم المطوّرون رد فعل سريع على طلبات الزبون وبشكل دائم
- **البساطة:** يجب أن يحافظ المطورون على المبدأ (KISS (Keep It Simple and Stupid
- **رد الفعل:** يجب أن يقوم المطورون بتعديلات تزايدية incremental حتى يكبر النظام تدريجياً، كما يجب أن يستوعبوا التعديلات ويعملوا على احتوائها لا مجرد أن يقبلوا بها.
- **الشجاعة والذهنية المتوقدة** التي يجب أن يتحلى بها المطورون.

كما تعتمد XP المبادئ الأساسية التالية لصنع أنظمة ناجحة:

- الاختبارات المستمرة
- الترميز البسيط الذي يقوم به زوجان من المبرمجين
- التواصل الوثيق مع المستخدم لبناء النظم بناء سريعاً

وبعد عملية تخطيط سريعة، تبدأ فرق العمل بإجراء التحليل والتصميم والتنجيز بشكل تكراري (انظر الشكل(6)).



الشكل (6) المنهجية المعتمدة البرمجة الحدية

اختيار منهجية التطوير المناسبة

ليس من السهل اختيار المنهجية المناسبة للتطوير، إذ لا توجد منهجية أجمع المطورون على أنها هي الفضلى!

كما أن لكل شركة تطوير معاييرها ومقاييسها النموذجية.

سنسلط الضوء في هذه الفقرة على بعض النقاط التي يمكننا استخدامها كمعايير للمقارنة بين المنهجيات.

❶ وضوح متطلبات المستخدم:

عندما يقدم لك المستخدم متطلبات غير واضحة عن النظام وعما يجب أن يفعله، يكون من الصعب فهم هـ ذه المتطلبـ ات

بالحديث عنها أو بكتابة تقرير حولها. يحتاج المستخدم في هذه الحالة إلى التفاعل مع التقانة لفهم ما سيفعله النظام وكيف يمكن تطويع هذه التقانة لاحتياجات الزبون.

في مثل هذه الحالات تكون المنهجيات المعتمدة على النمذجة الأولية وعلى النمذجة الأولية مع رمي النموذج هي الأكثر مناسبة لأنها تقدم نماذج أولية للمستخدمين تمكنهم من التفاعل معها في مرحلة مبكرة من دورة الحياة

❷ التآلف مع التكنولوجيا:

إذا كان النظام مصمماً دون أن يكون فريق المشروع متآلفاً مع التكنولوجيا الأساسية فيه، تزداد المخاطرة لأن الأدوات قد لا تكون قادرة على فعل المطلوب منها. في هذه الحالة يكون استخدام المنهجيات المعتمدة على النمذجة الأولية مع رمي النموذج هو الخيار الأفضل، في حين لايكون استخدام النمذجة الأولية مناسباً.

❸ تعقيد النظام:

تحتاج النظم المعقدة إلى تحليل وتصميم دقيقان. يمكن في هذه النظم استخدام النمذجة الأولية مع رمي النموذج، أو استخدام المنهجيات المعتمدة على التصميم البنيوي. أما استخدام التطوير على مراحل، فقد بينت التجربة أن فرق العمل الذي اعتمدته كانت تولي تحليل النظام المعقد اهتماماً أقل مما لو استخدمت منهجيات أخرى.

❹ موثوقية النظام:

تعتبر موثوقية النظام عاملاً هاماً في تطوير النظم. وتشكل المنهجيات المعتمدة على النمذجة الأولية مع رمي النموذج الخيار الأفضل عندما تكون الموثوقية ذات أولوية عالية. أما استخدام النمذجة الأولية فلا ينصح به هنا لأنه تنقصه الدقة والتأني في مرحلتي التحليل والتصميم.

❺ الخطط الزمنية قصيرة:

تناسب هذه المشاريع المنهجيات المعتمدة على التطوير السريع للتطبيقاتRAD. كما أن النمذجة الأولية والتطوير على مراحل يشكلان خيارين ممتازين لمثل هذه المشاريع. أما التطوير الشلالي فهو الخيار الأسوأ ويجب الابتعاد عنه.

❻ متابعة الخطة الزمنية:

لا توفر جميع المنهجيات القدرة على متابعة الخطط الزمنية والتحقق من مدى التقيد بها بدرجة واحدة. ونظراً إلى أن التصميم البنيوي يترك التصميم والتنجيز للمراحل الأخيرة، فإنه يخشى من عدم التمكن من المتابعة. تنقل منهجيات RAD الكثير من قرارات التصميم إلى البدايات مما يسمح بالتعرف إلى مواطن المخاطرة العالية والتصدي لها مبكراً.

اختيار منهجية التطوير المناسبة

يلخص الجدول التالي هذه المعايير للمفاضلة بين المنهجيات:

	Structured Methodologies		RAD Methodologies			Agile Methodologies
Ability to develop System	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP
With Under User Requirements	poor	poor	Good	Excellent	Excellent	Excellent
With Unfamiliar Technology	poor	poor	Good	poor	Excellent	poor
That are complex	Good	Good	Good	poor	Excellent	poor
That are Reliable	Good	Good	Good	poor	Excellent	Good
With a short time schedule	poor	Good	Excellent	Excellent	Good	Excellent
With schedule visibility	poor	poor	Excellent	Excellent	Good	Good

جدول المصطلحات:

ability to develop systems	إمكانية تطوير النظم	structured methodologies	المنهجيات البنيوية
RAD methodologies	منهجيات التطوير السريع	Agile methodologies	المنهجيات الرشيقة
Waterfall	شلالية	Parallel	على التوازي
Phased	على مراحل	Prototyping	النمذجة الأولية
Throwaway Prototyping	النمذجة الأولية مع رمي النموذج	XP	البرمجة الحدية
Excellent	ممتازة	Poor	ضعيفة
Good	جيدة	With unclear User requirements	مع متطلبات المستخدم غير واضحة
With unfamiliar Technology	مع تكنولوجيا غير مألوفة	that are complex	التي تتميز بالتعقيد
that are reliable	التي يجب أن تكون موثوقة	with a short time schedule	ضمن فترة زمنية قصيرة
with schedule visibility	متابعة الخطة الزمنية		

مهارات فريق التطوير وأدوار العاملين فيه

لضمان نجاح المشروع ، يجب أن يتألف فريق التطوير من مجموعة من العناصر الذين يمتلكون مهـ ـارات مختلفـ ـة. إن عناصر فريق المشروع هم عناصر التغيير، فهم الذين يبحثون عن طرق تحسين المؤسسة وهـ ـم الـ ـذين يبنـ ـون النظـ ـام البرمجي الذي يدعم جهودهم، وهم الذين يحفزون الآخرين ويدربونهم على استخدام النظام.

وإن نجاحهم في مهامهم هذه يضمن نجاح المشروع. وعليه، فقد جرى تحديد المهارات التي يجب أن يمتلكها المحللون في مجموعة من المهارات سنستعرضها على التتالي.

مهارات تقنية:	ليتمكن من فهم البيئة التقنية المتوفرة لدى المؤسسة والأسس التقنية للنظام الجديد والطريقة التي تسمح بالانتقال إلى الحل التقني المقترح.
مهارات في مجال الأعمال:	ليتمكن من فهم كيف يمكن تطبيق تقانات المعلومات على الأعمال والقيم المضافة التي تنتج عن ذلك.
مهارات تحليلية:	يحتاجها المحلل بشكل دائم عند كل مشكلة تحتاج إلى حل على الصعيد التنظيمي وضمن المشروع
مهارات تفاعلية:	يحتاجها المحلل أثناء تفاعله مع المستخدمين والمدراء والمبرمجين
هارات إدارية:	ملإدارة العاملين وإدارة الضغوط والمخاطر الناتجة عن التوصيف غير الواضح
مهارات أخلاقية:	بالطبع يجب أن يكون المحلل منصفاً وصادقاً وأخلاقياً مع جميع من حوله. كما يجب عليه أن يحترم خصوصيات المؤسسة التي يبني لها النظام وسريتها وخصوصية معلوماتها.

إضافة لهذه المهارات العامة، فإن الدور الذي يسند إلى كل محلل يتطلب منه أن يكون حائزاً علـ ـى المهـ ـارات الخاصـ ـة اللازمة له. نبين فيما يلي الأدوار التي يمكن أن تسند إلى المحلل:

- محلل للأعمال Business Analyst
- محلل للنظام Systems Analyst
- محلل للبنية الأساسية Infrastructure Analyst
- محلل لإدارة التغيير Change Management Analyst
- مدير للمشروع Project Manager

تدريب (1):

حدد منهجية التطوير المناسبة لتطوير النظم التالية:

1) نظام مراقبة وضع المريض في غرفة العناية المشددة

2) نظام ذاتية العاملين في الجامعة الافتراضية (بديل عن نظام موجود سابق)

3) نظام القيادة الآلية لطائرة.

4) نظام مساعد على اتخاذ القرار يجمع معلوماته من قواعد معطيات موجودة ويقدمها للمدراء.

5) نظام لإصدار البطاقات يستخدمه المسافرون في محطة القطار

6) نظام حضور المؤتمرات عن بعد video conferencing يسمح بعرض الفيديو والصوت لعدة مشاهدين في نفس الوقت

7) ربوط ينظف الأرضيات مصنوع لتنظيف الأماكن ذات المساحات الواسعة والتي لاتوجد فيها حواجز كالممرات. يستطيع الربوط أن يتحسس الجدران والعقبات الأخرى.

تدريب (2):

قارن بين تطوير البرمجيات باستخدام البرمجة الحدية والنمذجة الأولية مع رمي النموذج.

تدريب (3):

قارن بين تطوير البرمجيات على التوازي وتطويرها على مراحل مع التركيز على محاسن ومساوئ كل من المنهجيد ين

نهاية الوحدة